The Robot Task Space Analyzer

William R. Hamel (whamel@utk.edu, (423)-974-6588)
Mechanical Engineering
207 Dougherty Engineering Building
University of Tennessee
Knoxville, Tennessee 37996-2210

ABSTRACT

A promising way to achieve increased remote worksystem efficiency is to layer telerobotic technologies onto teleoperated remote systems. The research being reported here enables the teleoperation baseline to be supplemented with operator-selective telerobotic modes of operation that allow automatic performance of subtasks that are either repetitive, require high precision, or involve extreme patience. Before subtask automation can be exploited, however, it is necessary to explicitly represent the 3-D geometry of the task space scene surrounding the remote worksystem. The Robot Task Space Analyzer (RSTA) is a tool for remote equipment operators that combines infrared laser and visible stereo imaging, human-interactive modeling and computer-based object recognition to build 3-D models of the immediate work zone in which a robot system is operating. Ultimately, this model will be used by the telerobot control system in automatic collision checking and motion planning routines so that some aspects of the remote tasks can be performed robotically. This paper presents the hardware and software design of the RTSA system. It also discusses results of preliminary laboratory testing which was performed to evaluate the model building time efficiency and model accuracy. Human factors aspects the system operation and design are discussed. Plans for full-scale testing in DOE facilities are summarized.

1. Introduction

Environmental restoration and waste management (ER&WM) challenges in the United States, and around the world, involve radiation or other hazards which will necessitate the use of remote operations to protect human workers from dangerous exposures. Remote operations carry the implication of greater costs since remote work systems are inherently less productive than contact human work due to the inefficiencies/complexities of teleoperation. To reduce costs and improve quality, much attention has been focused on methods to improve the productivity of combined human operator/remote equipment systems; the achievements to date are modest at best. The most promising avenue in the near term is to supplement conventional remote work systems with robotic planning and control techniques borrowed from manufacturing and other domains where robotic automation has been used. Practical combinations of teleoperation and robotic control will yield telerobotic work systems that outperform currently available remote equipment. It is important to recognize that the basic hardware and software features of most modern remote manipulation systems can readily accommodate the functionality required for telerobotics. Further, several of the additional system ingredients necessary to implement telerobotic control – machine vision, 3D object and workspace modeling, automatic tool path generation and collision-free trajectory planning – are existent.

Practical and reliable implementation of telerobotic systems in ER&WM contexts is an unrealized objective, despite the potential payoff of telerobotics. This can be attributed to several formidable technical challenges unique to field automation. Almost always the geometry of the task environment is highly unstructured and uncertain. Likewise, the precision and accuracy of the requisite geometric knowledge varies from task to task, as does the extent of the task space itself. A significant fraction of the tasks to be performed are complex by any standard. These factors put full automation of ER&WM tasks beyond the reach of current technology. However, there are certain subtasks that are amenable to automatic planning and execution by interjecting telerobotic subtasks into the overall sequence. Implementation of telerobotic capability in a typical ER&WM application will involve operational sequences such as that as depicted in Figure 1.

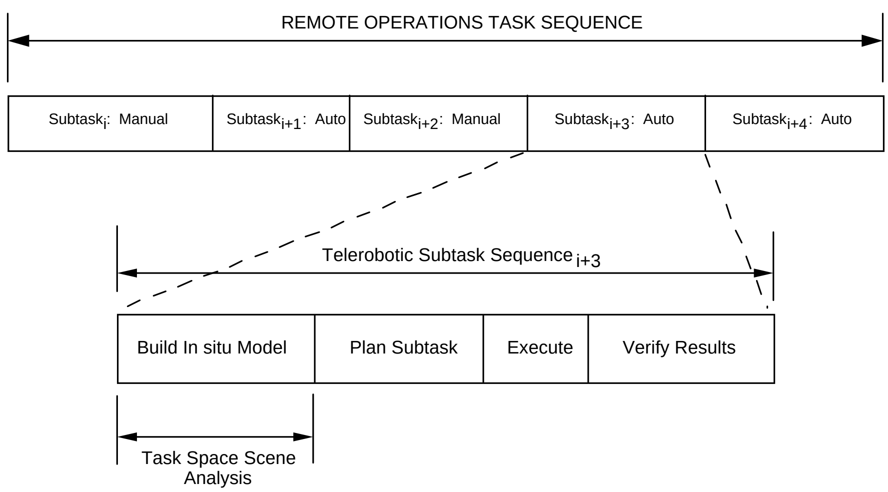


Figure 1, Telerobotics Operations Cycle

The type of operation implied by Figure 1 puts emphasis on the human-machine interaction and cooperation. In the case of RTSA, it is believed that human-interactivity is foundational for ultimate task space modeling efficiency as well as seamless maneuvering between manual and automated operations.

2. Robot Task Space Analyzer Concept

Automation of a task requires complete quantitative data about the task/subtasks to be performed, the manipulation systems, and the tooling devices to be used. Task space scene analysis (TSSA) refers to the process by which the remote work system gathers geometrical and other types of information that are necessary to characterize, analyze, and plan the automated task execution [1,2]. For example, in a dismantlement scenario the task may be to remove a segment of process piping using remote manipulators and cutting tools. If such a task is to be automated, it is necessary to describe the location and orientation of each piping element with respect to the remote work system. This data representation, or model, must be complete and accurate to an extent dictated by the specific tool being used: positioning of a shear demands less accuracy than maintaining the proper standoff for a plasma arc torch. Once a sufficient model is available, planning the manipulator and tooling motions can be defined, and the cutting can be automatically executed. The RTSA is a system that performs TSSA, and is in essence a model builder of the near field of view of the mobile work system. Unlike the notion of world model building, RTSA functions in the region of "space" in the near field that is within the sphere of influence of the remote work system where the current task operations are to be performed. RTSA performs an integral step in the telerobotics operations cycle and it must exhibit a level of efficiency that allows telerobotic execution to provide performance benefits over conventional teleoperational execution.

As depicted in Figure 1, telerobotic execution requires a "programming" phase and an "execution" phase for each task to be performed. The programming phase is the RTSA function plus task planning; it is the most important part of the operation since subsequent execution is fully automatic and can progress at the full operating speed of the remote hardware. Therefore, RTSA is an enabling technology that determines the ultimate overall performance of any telerobotics concept.

2.1 Functional Architecture

The RTSA has three major components as seen in Figure 2. The components are based on the work previously done with Human-Interactive Stereo [1,2], Artisan [3], and a manual model building component where the operator input is used exclusively. From a panoramic view (PV) of the task scene, the operator selects a region of interest (ROI) and assigns the building pipes and fittings models in the ROI to be done manually or to be done by an AutoScan method (an automated method). When the user chooses to have a ROI analyzed automatically, the stereo AutoScan or the range AutoScan function would be chosen along with specified classes of objects to be found in the ROI. In its current implementation, RTSA contains object classes for standardized process piping components and a custom object tool. The class of objects describes the schedule and size of the piping and whether it is welded, flanged, or screwed piping; this also includes the fitting or fittings to be found including tees, elbows, and pipes.

Human-interactive stereo [1,2] used a pair of black and white cameras to capture images of the task scene. Once the images were displayed, the operator indicated corresponding points of a pipe segment in each image. From points at each end of the pipe segment, stereo calculations could be made to construct a 3D model of the pipe including its size, position, and orientation.

CMU's Artisan [3,4] is a perception system that automatically creates three dimensional models of the area in which a robot works. An operator begins a session with Artisan by instructing the system to acquire range data of the scene using a scanning laser range finder or structured light sensor. Special filtering algorithms are applied to the range image to further reduce noise (while preserving the range discontinuities) and the images are displayed on the operator's workstation. Since the sensor field of view is usually larger than the area the operator wishes to work on, he restricts the system's attention to a particular region of interest by drawing a box around it. Next, he indicates what objects Artisan should expect to find in the region of interest by selecting from a menu of pre-defined object types and sizes. Artisan then creates a Cartesian mesh from the range data in the region of interest thus defining a 3-D surface representation of the data.

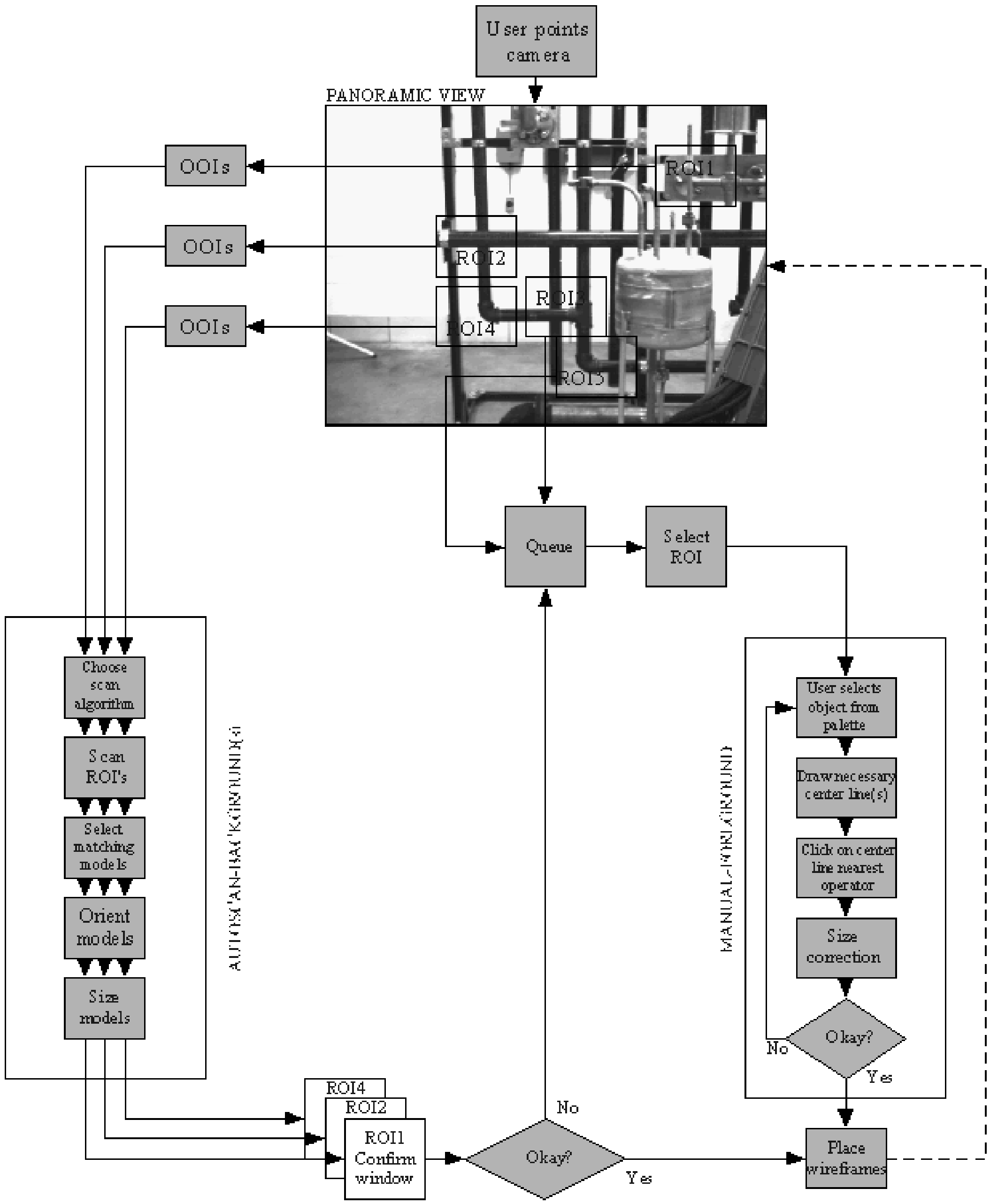


Figure 5, RTSA Functional Architecture

Two different object recognition algorithms have been developed for Artisan. The first method (Quadric/Planar Segmentation and Matching, or "QPSM") segments the 3-D surfaces into planar and quadric patches and matches the resulting scene description to analogous descriptions of object models in a database (developed off-line from CAD descriptions of objects). The other method (Free-form Object Recognition Method or "FORM") is based on a technique known as geometric indexing. In this case a collection of 3-D surface points is transformed into a set of 2-D representations, called spin images, that describe the spatial relationship of each point to all the others. The stack of spin images representing the scene data are then compared to stacks of spin images of models in the database to arrive at a few number of plausible correspondences. Each of these is further refined using a modified iterative closest point (ICP) algorithm that outputs the optimal estimate of the recognized object's dimensions, location and orientation in the task space. For each object recognized, the operator can either accept or reject what Artisan found in the data. Each accepted object appears in the World Model window in the location that Artisan has calculated. This process of range data collection, processing and user interaction continues until the operator is satisfied with the 3-D model of the robot's work space.

By selecting ROIs, the operator limits the volume of information required to be analyzed by either of the background AutoScan algorithms and increases their collective efficiency. While the AutoScan algorithms are being executed in the background, the operator can build models of the pipes/fittings manually in other ROIs in the foreground. The operator's list of ROI's assigned to be analyzed manually is known as the manual queue.

The structure of the RTSA flows naturally from the desire to automatically develop models with the AutoScan methods and the need to have operator input. With three paths available for the creation of the task space model, the operator is both an administrator and an active participant.

Administratively, the operator separates the scene into ROIs and assigns the ROIs to be sent either to the manual queue or to an AutoScan method. By allowing the operator to assign parts of the scene to an AutoScan method, the operator's knowledge of an AutoScan method's past successes and failures will aid in the his decision to use AutoScan. Under certain scene conditions, such as occlusions and poor lighting, the operator can decide which method to use on specific regions. During manual modeling the operator designates the placement of the object with the laser range pointer and then approves the object placement by making small adjustments in translation and rotation of the on screen model as is done when the operator approves the results of an AutoScan algorithm.

The operator's input, in the form of manually placing objects in a ROI, is essential. The operator's skill for recognizing objects in a ROI as well as the intuitive ability to place and orient those objects makes him the most robust avenue to creating a model of the scene. The operator also acts as backup to the AutoScan methods as each object can be tweaked into the correct position and orientation if the AutoScan method does not produce modeling results of sufficient accuracy. In the event that an AutoScan method fails by missing an object or by placing an erroneous object, the operator can complete a partially modeled ROI or delete those objects that don't belong. By displaying a visual representation of the model in front of the stereo images, the operator can approve or disapprove of the model built by an AutoScan method.

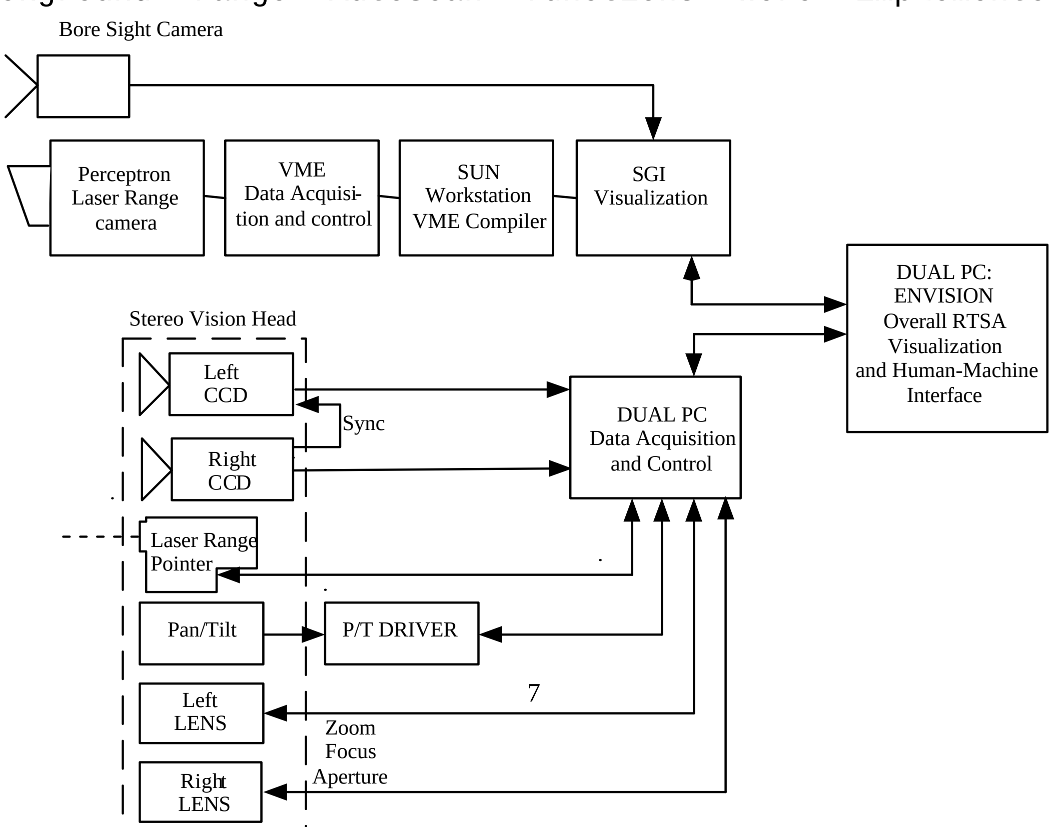
The stereo AutoScan algorithm uses a pair of images taken from a set of black and white, charge coupled device (CCD) cameras with servo lenses mounted on a pan-tilt head. The stereo head points to the appropriate ROI and acquires a set of stereo images. The stereo images are supplied to the stereo AutoScan algorithm with the desired class of objects to be found. Unlike the previous work done in Human-Interactive Stereo, a model of the class of objects already exists so certain parameters such as pipe diameter and elbow radius are already known. Standard piping and fittings for various pressure ratings and line sizes have been included in an object library within RTSA. Automated object recognition and positioning is greatly simplified with the limitation to the class of standard piping. The algorithm finds the location and orientation of the objects of interest (OOI) in the task scene such as pipes, elbows, and tees.

2.2System Implementation

The RTSA implementation philosophy is intended to reduce the ultimate recurring costs of systems by maximizing the use of low-cost PC-based software and hardware.

2.1.1 Hardware

The overall hardware architecture is shown in Figure 3. Initially, the background range AutoScan functions were implemented on a



separate Silicon Graphics Workstation in order to

Figure 3, RTSA Hardware Configuration

satisfy budget and schedule constraints. In the near future, all of the foreground and background modeling functions will be implemented within the two dual-PC workstations.

The computer controlling the stereo head is a Dell 400 workstation with dual 300MHz Pentium II™ processors. The intensive video processing associated with RTSA is performed by an Elsa™ Gloria-XL video card. The stereo head is controlled through a set of four serial ports; one serial port was required for each of the servo lenses, one for the pan and tilt drives, and one for the laser range pointer. The images are acquired through a Matrox™ Meteor RGB/PPB frame grabber. An RGB frame grabber was chosen so that the black and white images could be captured on different channels – red, green, or blue – of the frame grabber with inherent synchronization. The other computer used is a Dell 400 workstation as well and uses dual 333MHz processors. This computer is used for software development and stereo AutoScan algorithm execution.

2.2.1 Software

The operating system chosen is Windows NT™; NT was a natural choice for an operating system as it is a low-cost, widely used, and stable operating system. Operating systems for robotics applications have included products such as VxWorks™ and QNX™ to provide real time operation. RTSA, however, doses not require precise real-time execution since it performs “off-line” modeling functions rather than control. An added bonus with NT is the simplicity of hardware integration; specifically, the drivers for the frame grabber were available. Additionally, there is a clear trend of the expanding use of NT in engineering embedded applications beyond computer networking. Given the operating system and experience with the C language, the choice of development tools used to write the program is Microsoft Visual C++™ 5.0. The Microsoft developer's environment was found to be an effective program development environment for this application.

2.2.2 3D Model Display

One of the most critically important aspects of RTSA is to provide the operator (and the computer model) with an effective 3D model representation and visualization medium. In the interest of time and risk, a commercially available 3D package was chosen for use with RTSA instead of creating a custom modeling environment. The package that was chosen was the Deneb product called Envision™ VP. Envision is a 3D kinematic modeling package most often used in simulating the motion of manipulators and virtual path planning. Models of the pipe fittings were made in Envision.

2.3 Sensor Configurations

The stereo sensor head consists of two Newport™ drives and stepping motors in a pan/tilt arrangement, two Panasonic™ CCD cameras, two Electonique-Informatique Applications™ (EIA) servo lenses, and a SICK™ laser range pointer. The Newport drives and stepper motors allow sensor head pointing with a step size of one one-thousandth of a degree. The CCD cameras are Panasonic GP-MF552 units that produce black and white images with 640 X 480 resolution. The EIA servo lenses, Model X6, allow the digital control of focus, zoom, and aperture. The laser range pointer is a SICK model DME 2000 that measures the phase of a returning laser beam to determine distance to a reflecting surface.

As seen in Figure 4, the drive at the base of the sensor head is in the pan drive and is located in the horizontal plane. The tilt drive is mounted above the pan drive so that the vertical plane in which the tilt drive operates is rotated by the pan drive. There are two brackets attached to the tilt drive that hold the CCD cameras. Under the left bracket as viewed from behind the sensor head and facing the mockup is the laser range pointer. The laser range pointer is mounted as close to the tilt drive as possible to minimize any deflection in the bracket that the added weight of the laser range pointer might induce. The EIA servo lenses and CCD cameras are mounted under the ends of the brackets

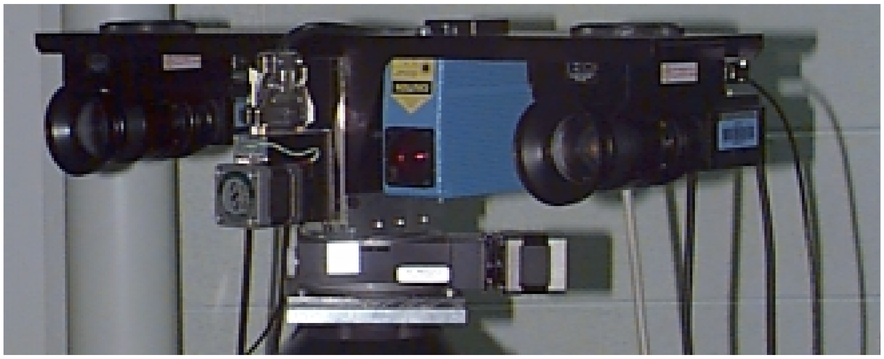


Figure 4, Stereo Sensor Head

The laser range camera used in initial RTSA is the Minolta Vivid 7000 laser range camera, but other systems are being considered for full-scale testing. The range camera has 8X zoom and a maximum field of view of thirty degrees in its zoomed out configuration. The range of the camera is 600 to 3000 millimeters. The Minolta uses a structured light approach to calculate the distance to the object by scanning the scene with a laser and recording the location of the light with a sensor mounted at a known distance from the source. By recording the location in the sensor image and knowing the off-set from the source, a distance to the scene object can be calculated. The Minolta Vivid 7000 is capable of determining the distance of objects in one field of view to millimeter accuracy in 3 seconds and coregisters a pseudo-color image with the range image. For communications with a computer, the range camera uses Small Computer Systems Interface(SCSI).

3. RTSA Graphical User Interface

The graphical user interface (GUI) is the "connection" between the computer and the operator and is one of the most important aspects of the system. All of the operator input required by RTSA goes through the GUI, and all the information required by the operator is displayed by the GUI. A successful computer-based system is one that allows the operator to get information from and supply information to the computer in a natural way; this natural flow of information requires the GUI to present information in an intuitively obvious manner. The study of the flow of information with a computer is a key area of human factors engineering. The human factors variables that were controllable in the RTSA GUI were manipulated to make it as user friendly as possible. If RTSA is less easy to use than to control the manipulators under teleoperation control, the operator will most likely choose to complete all the tasks in teleoperation mode. In the following discussion, the flow of information at the GUI to and from RTSA are discussed. The GUI windows and their hierarchy are shown in Figure 5.

3.1 Defining Regions of Interest

After calibrating the sensors and obtaining the desired panoramic view of the task space of interest, the operators task is to subdivide the task analysis into region of interest (ROI) that contain objects that must be modeled. Refer to step 3 in Figure 4. The object of splitting the task scene PV into ROIs is to allow different processes to work on different parts of the scene at the same time. The ROIs are a way for the operator to keep track of which modeling method is being executed in which part of the scene. Also, ROIs speed up the AutoScan methods, i.e., each ROI can be analyzed more quickly than the entire scene. Duplication of efforts that would occur, if both AutoScan methods were used to analyze the same object, is not a concern.

3.2 Placing Objects Manually

The manual placement of object models must be the most intuitive and practical part of the program as it will determine if a task scene is modeled in a timely matter. Refer to step 4a in Figure 4. The position and orientation of each object model must be designated by the operator using the laser pointer along with the dimensions of the model being placed. For the pipe, the orientation around the pipe axis does not matter and the length can vary; so two points in space (as defined with the laser pointer) can accurately define the end points of a pipe and with the diameter information from the model fully define the pipe. In the case of an elbow, the dimensions of an elbow are known, but all orientation axes must be defined requiring three points. As for the tee, like the elbow the tee is of fixed dimensions and requires three points (from the laser pointer) to describe its orientation.

3.3 Choosing Object Class Information

The definition of the object class information is necessary for the AutoScan methods. Refer to step 4b in Figure 4. This information is necessary because the AutoScan methods are model-based; so a correct mode representation of the OOI must be supplied to the algorithms. For example, since the RTSA operator would know the difference between a three inch pipe and a two inch pipe, the AutoScan methods need not waste time attempting to determine the size pipe that is not in the scene and can eliminate from consideration OOIs that appear to be of diameters different from that specified by the operator.

3.4 Information Required from the RTSA

As mentioned earlier, the result of the program is the model of a task space scene. For a model of the task space scene to be built, the operator needs to be presented with pertinent information from the RTSA. For example, in the placement of points (i.e., the laser pointer spots) when defining the location of objects manually, the operator needs an interactive screen to zoom in on the OOI and position the laser range finder dot on that object. Also, the validation of the correct placement of object models from either the manual identification of points or the AutoScan algorithms requires a view of the task space scene. This task space scene view needs to incorporate the object models' placement information so the operator can visually inspect the object models' placement in comparison to the actual location of the objects. The last example is that the ROI information needs to be presented to the operator so that effort is not wasted on modeling an area that has already been modeled or on assigning an area to be modeled by two methods. By supplying the operator with pertinent information, the operator's efficiency in modeling the task space scene can be maximized. In its current form, the RTSA GUI involves seven separate windows and requires no keyboard input if the mouse is used to control the stereo sensor head pan and tilt motions. Work continues on the evaluation and streamlining of the GUI. The goal is eliminate all keyboard operations and to minimize the number of windows while keeping their structures simple.

4. Experimental Evaluations

As discussed at the outset, the quantitative performance of the RTSA process is critically important with regard to the practicality of remote telerobotics. In the interest of quantitative evaluation, a structured modeling environment and experimental scheme has been developed.

4.1 Test Mock-Up

The task space scenes shown in Figure 5 show the task space mock-up that was constructed for RTSA testing. The mock-up provides the density and size of process piping objects that one would expect in a typical task space scene. The mock-up was constructed from conventional piping components including some stainless steel items. Image properties such as occlusions, surface colors, and surface spectral characteristics are realistic. A precise Envision 3D graphical model (± 0.25 inches) of the mock-up was constructed and is used to compare RTSA modeling results with “ground truth.” The graphical model is “calibrated” relative to the true mock-up position using a theodolite with range measurement capability (i.e., Hewlett Packard Total Station™). The Total Station is also used to establish the coordinates of the sensor head relative to the task mock-up. Coordinate transformations were developed to allow the RTSA modeling results and the graphical model to be expressed in terms of a coordinate frame located at the base position of the actual mock-up. This allows RTSA results to be superimposed with the graphical model to provide an excellent visualization of the model correspondence with the real world. The standoff distance between the sensor head tripod and the task mock-up was approximately 16 feet.

In addition, human factors test principles were developed to evaluate operator performance, identify areas of difficulty, and to record erroneous operations. Observations of several operators (primarily graduate students) were performed as they attempted to model the lower left corner of the mock-up.

4.2 Testing

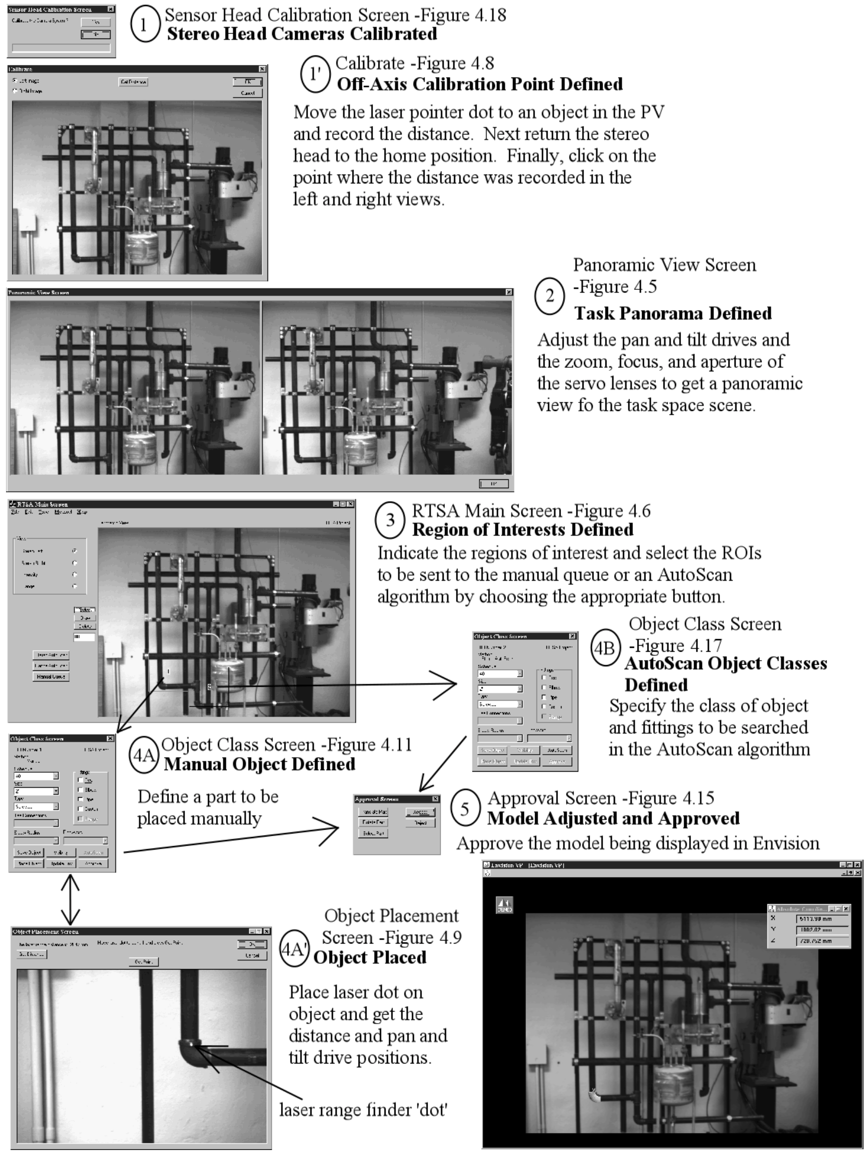


Figure 5, RTSA Operational Flow

At the time of initial testing, the RTSA software was not completely debugged and robust. Nonetheless, essentially all of the key features of the RTSA operational flow and modeling accuracy were studied for straight pipe connected with tees and elbows. Complete modeling results were obtained for foreground manual modeling only at this stage. At the time of the initial testing the background AutoScan functions were not fully operational. In the D&D context used to guide this research, the removal of the lower left corner section might be a “chunk” that an operator would consider assigning as a typical automated subtask. This section contains one elbow, one tee, and five straight sections of pipe.

The goal of testing was to determine the speed and accuracy with which a typical task could be manually modeled. Timing data between novice and expert users was compared to determine the ease of using the interface, and also comparisons between joystick and mouse control of the pan/tilt.

Subjects

The novice group consisted of 3 graduate students from the psychology department and the expert group consisted of 2 mechanical engineering students and 1 mechanical engineering faculty member who had experience with the RTSA program.

Procedure

The modeling task consisted of placing a seven-item section from the mockup, which included one (1) tee, one (1) elbow, and five (5) pipes. Subjects were given the opportunity to practice modeling each type of object with RTSA until they felt comfortable with their ability to understand and use the system. They were also given a choice of using the joystick or mouse controls, and allowed to try each beforehand.

Each subject modeled the test section of the mockup twice. The start-point in the mockup (where modeling began) was randomly selected by the experimenter. Subjects were instructed to model the given section as quickly and accurately as possible, and to use the translation and rotation features of RTSA to adjust any parts they thought needed it.

The data collected were based on the location error and time to complete the task. The errors in locating each component were recorded from Envision for 1) where the RTSA manual modeling initially placed the component, and 2) the location of the part after being adjusted by the subject. The time was recorded to complete the initial placement and adjustment phases for each component.

Results

When given a choice of using the joystick or mouse, all subjects preferred using the joystick, with some using the mouse for fine-tuning and smaller adjustments.

<u>Accuracy</u>: Average error in the initial placement of a part was 65.6 mm (s=32.52mm), representing an error rate of 5-7% of the entire distance for any given dimension (x, y, or z) of a part. Error for each part and dimension, along with relevant interactions, are presented in the figures below.

Error was significantly greater in the Y dimension for novices. (F=11.877, p=.00). Error in the Z dimension was significantly greater for tee in both novice and expert groups.(F=22.457, p=.00).

Adjustment did not always improve accuracy, and in the case of our subjects increased error by an average of 9.87 mm.

Timing: The average time to completely model the test section with RTSA, including adjustment, was 6.6 minutes (396 seconds). Without adjustment, the average time for modeling the section was 5.06 minutes (303.5 seconds). Both novices and experts showed improvement with practice, with novices making the largest improvement and reaching performance levels of the experts by the second time through (novices from 500 to 343 seconds, and experts 405 to 345 seconds).

Discussion

Overall, the results demonstrate the lack of consistent overall differences. All of the significant effects occur in the interaction between pairs of variables, particularly the interaction between the experience of the subject and the size of the X,Y,Z error and the interaction between the type of component and size of the X,Y, Z error.

First, the major difference between novices and experts mainly occurs in the Y dimension for all component types. Although performance was generally better for the experts, for the Y